

Search Results

BROWSE

SEARCH

IEEE XPLORE GUIDE

SUPPORT

Results for "(((multi-processor or multiprocessor) and ((shared memory) or (shared-memory))) and (non-block or loc...)"

e-mail printer friendly

Your search matched **14** of **1476571** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

[View Session History](#)

[New Search](#)

» Key

IEEE JNL IEEE Journal or Magazine  
IEE JNL IEE Journal or Magazine  
IEEE CNF IEEE Conference Proceeding  
IEE CNF IEE Conference Proceeding  
IEEE STD IEEE Standard

Modify Search

☐ Check to search only within this results set

Display Format: ☒ Citation ☐ Citation & Abstract

[Select All](#) [Deselect All](#)

- ☐ 1. **Managing concurrent access for shared memory active messages**  
Lumetta, S.S.; Culler, D.E.;  
[Parallel Processing Symposium, 1998. 1998 IPPS/SPDP. Proceedings of the First Merged International...and Symposium on Parallel and Distributed Processing 1998](#)  
30 March-3 April 1998 Page(s):272 - 278  
Digital Object Identifier 10.1109/IPPS.1998.669925  
[AbstractPlus](#) | Full Text: [PDF](#)(780 KB) IEEE CNF  
[Rights and Permissions](#)
- ☐ 2. **Relative performance of preemption-safe locking and non-blocking synchronization on multiprogrammed shared memory multiprocessors**  
Michael, M.M.; Scott, M.L.;  
[Parallel Processing Symposium, 1997. Proceedings., 11th International](#)  
1-5 April 1997 Page(s):267 - 273  
Digital Object Identifier 10.1109/IPPS.1997.580906  
[AbstractPlus](#) | Full Text: [PDF](#)(696 KB) IEEE CNF  
[Rights and Permissions](#)
- ☐ 3. **Coherence communication prediction in shared-memory multiprocessors**  
Kaxiras, S.; Young, C.;  
[High-Performance Computer Architecture, 2000. HPCA-6. Proceedings. Sixth International Symposium on](#)  
8-12 Jan. 2000 Page(s):156 - 167  
Digital Object Identifier 10.1109/HPCA.2000.824347  
[AbstractPlus](#) | Full Text: [PDF](#)(244 KB) IEEE CNF  
[Rights and Permissions](#)
- ☐ 4. **Towards a practical snapshot algorithm**  
Riany, Y.; Shavit, N.; Touitou, D.;  
[Theory of Computing and Systems, 1995. Proceedings., Third Israel Symposium on the](#)  
4-6 Jan. 1995 Page(s):121 - 129  
Digital Object Identifier 10.1109/ISTCS.1995.377039  
[AbstractPlus](#) | Full Text: [PDF](#)(712 KB) IEEE CNF  
[Rights and Permissions](#)
- ☐ 5. **Reactive multiword synchronization for multiprocessors**  
Ha, P.H.; Tsigas, P.;  
[Parallel Architectures and Compilation Techniques, 2003. PACT 2003. Proceedings. 12th International Conference on](#)  
27 Sept.-1 Oct. 2003 Page(s):184 - 193  
Digital Object Identifier 10.1109/PACT.2003.1238014  
[AbstractPlus](#) | Full Text: [PDF](#)(499 KB) IEEE CNF  
[Rights and Permissions](#)

- ┌ **6. The impact of instruction-level parallelism on multiprocessor performance and simulation methodology**  
 Pai, V.S.; Ranganathan, P.; Adve, S.V.;  
High-Performance Computer Architecture, 1997., Third International Symposium on  
 1-5 Feb. 1997 Page(s):72 - 83  
 Digital Object Identifier 10.1109/HPCA.1997.569611  
[AbstractPlus](#) | Full Text: [PDF\(1248 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ┌ **7. Practical considerations for non-blocking concurrent objects**  
 Bershad, B.N.;  
Distributed Computing Systems, 1993., Proceedings the 13th International Conference on  
 25-28 May 1993 Page(s):264 - 273  
 Digital Object Identifier 10.1109/ICDCS.1993.287700  
[AbstractPlus](#) | Full Text: [PDF\(848 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ┌ **8. Lock-free garbage collection for multiprocessors**  
 Herlihy, M.P.; Moss, J.E.B.;  
Parallel and Distributed Systems, IEEE Transactions on  
 Volume 3, Issue 3, May 1992 Page(s):304 - 311  
 Digital Object Identifier 10.1109/71.139204  
[AbstractPlus](#) | Full Text: [PDF\(732 KB\)](#) IEEE JNL  
[Rights and Permissions](#)
- ┌ **9. Lock-free scheduling of logical processes in parallel simulation**  
 Liu, J.; Nicol, D.M.; King Tan;  
Parallel and Distributed Simulation, 2001. Proceedings. 15th Workshop on  
 15-18 May 2001 Page(s):22 - 31  
 Digital Object Identifier 10.1109/PADS.2001.924618  
[AbstractPlus](#) | Full Text: [PDF\(844 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ┌ **10. Energy reduction in multiprocessor systems using transactional memory**  
 Tali Moreshet; Bahar, R.I.; Herlihy, M.;  
Low Power Electronics and Design, 2005. ISLPED '05. Proceedings of the 2005 International Symposium on  
 8-10 Aug. 2005 Page(s):331 - 334  
[AbstractPlus](#) | Full Text: [PDF\(309 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ┌ **11. Modeling and measuring hot spots on MIN-based and HR-based shared-memory architectures**  
 Zhang, X.; Yan, Y.; Castaneda, R.;  
Parallel and Distributed Processing, 1993. Proceedings of the Fifth IEEE Symposium on  
 1-4 Dec. 1993 Page(s):176 - 183  
 Digital Object Identifier 10.1109/SPDP.1993.395534  
[AbstractPlus](#) | Full Text: [PDF\(600 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ┌ **12. The impacts of timing constraints on virtual channels multiplexing in interconnect networks**  
 Khonsari, A.; Ould-Khaoua, M.; Nayebi, A.; Sarbazi-azad, H.;  
Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International  
 10-12 April 2006 Page(s):8 pp.  
 Digital Object Identifier 10.1109/.2006.1629390  
[AbstractPlus](#) | Full Text: [PDF\(228 KB\)](#) IEEE CNF  
[Rights and Permissions](#)
- ┌ **13. Fast synchronization on scalable cache-coherent multiprocessors using hybrid primitives**  
 Nikolopoulos, D.S.; Papatheodorou, T.S.;  
Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International  
 1-5 May 2000 Page(s):711 - 719  
 Digital Object Identifier 10.1109/IPDPS.2000.846056

14. **A loop-free asynchronous data sharing mechanism in multiprocessor real-time systems based on timing properties**

Jing Chen;

[Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on](#)

19-22 May 2003 Page(s):184 - 190

Digital Object Identifier 10.1109/ICDCSW.2003.1203552

[AbstractPlus](#) | Full Text: [PDF\(496 KB\)](#) IEEE CNF

[Rights and Permissions](#)

[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)

© Copyright 2006 IEEE – All Rights Reserved

## Full Citation & Abstract

PRICING ?

PUBLICATION AAT 3072778  
NUMBER

TITLE Speculation-based techniques for transactional lock-free execution of lock-based programs

AUTHOR Rajwar, Ravi

DEGREE PhD

SCHOOL THE UNIVERSITY OF WISCONSIN - MADISON

DATE 2002

PAGES 286

ADVISER [Goodman, James R.](#)

ISBN 978-0-493-92677-3

SOURCE DAI-B 63/11, p. 5340, May 2003

SUBJECT COMPUTER SCIENCE (0984); ENGINEERING, ELECTRONICS AND ELECTRICAL (0544)

DIGITAL FORMATS 9.88Mb image-only PDF

ORDER NOW ▶

FREE DOWNLOAD

24 page Preview



This thesis is motivated by the difficulty in writing correct high-performance programs. Writing shared-memory multithreaded programs imposes a complex trade-off between programming ease and performance, largely due to subtleties in coordinating access to shared data. To ensure correctness programmers often rely on conservative locking at the expense of performance. The resulting serialization of threads is a performance bottleneck. Locks also interact poorly with scheduling and faults. We seek to improve multithreaded programming trade-offs by providing architectural support for optimistic lock-free execution. In a lock-free execution, shared objects are never locked when accessed by various threads. We propose two hardware techniques: Speculative Lock Elision and Transactional Lock Removal. Speculative Lock Elision (SLE) is a micro-architectural technique to remove dynamically-unnecessary lock-induced serialization. The key insight is that locks don't always have to be acquired for a correct execution. Synchronization instructions are predicted as being unnecessary and elided. This allows multiple threads to concurrently execute critical sections protected by the same lock. Misspeculation due to inter-thread data conflicts is detected using existing cache mechanisms and rollback is used for recovery. Successful elision is validated and committed without acquiring the lock and non-conflicting critical sections execute and commit concurrently without serialization on the lock. SLE can be implemented entirely in the microarchitecture without instruction-set support and system-level modifications, is transparent to programmers, and requires trivial additional hardware support. Transactional Lock Removal (TLR) uses SLE as an enabling mechanism but in addition provides successful lock-free execution of lock-based critical sections in the presence of data conflicts if sufficient resources are available for buffering speculative state. TLR elides locks using SLE to construct an optimistic lock-free critical section execution but in addition also uses a timestamp-based conflict resolution scheme to provide lock-free execution even in the presence of data conflicts. By treating lock-free critical sections as lock-free transactions, TLR provides transactional properties to critical sections and by using timestamps also provides

starvation freedom. Benefits of SLE and TLR include improved programmability, stability, and performance. Programmers can obtain benefits of lock-free data structures, such as non-blocking and wait-free behavior, while using lock-protected critical sections for writing programs.